

ром 20; 40; 75; 100; 125; 150 мм одной длины 2.8 м. Прибор калибровался на трубе диаметром 20 мм и проводились измерения длин других труб. Затем прибор поочередно калибровался на трубах других диаметров и каждый раз проводились измерения их длин. График изменения погрешностей показан на рис. 4, из которого видно, что погрешности носят систематический характер. Таким образом, при измерении труб малого диаметра полученные корректирующие зависимости сохраняют свой вид и числовые значения. При измерении труб с отношением диаметра к длине, $L/d_K < 0.05$ коррекцию показаний необходимо проводить также по ранее полученным зависимостям, но окончательный результат получать путем введения дополнительного коэффициента, который может быть вычислен из графиков, показанных на рис. 3.

На основе проведенных исследований можно сформулировать следующий общий алгоритм функционирования прибора в различных вариантах использования.

При использовании в качестве уровнемера: прибор калибруется на трубе используемого диаметра длиной соответствующей максимуму диапазона преобразования. Вычисление длины производить по формуле (2) с корректирующей формулой (6) или (7) в зависимости от требуемой точности. При использовании в качестве трубомера возможно два пути:

1. Разделение всей номенклатуры труб на типоразмеры с диапазоном изменения относительных размеров гарантирующих не превышение систематических погрешностей допустимых значений. Прибор позволяет разбить диапазон всех измеряемых труб на 20 типоразмеров.

2. Прибор калибруется на максимальной длине среднего диаметра. На клавиатуре предусматривается возможность ввода параметра относительного размера L/d_K . Далее в зависимости от этого значения выбираются формулы вычисления (2), (6) при $L/d_K < 0.05$ или (2), (6), (8) $L/d_K > 0.05$.

Список использованных источников

1. Скворцов Б.В., Жиганов И.Ю. Электронно-акустические приборы измерения длины и расстояний с автоматическим зондированием свойств среды. // Тезисы доклада XI международной научно-технической конференции «Датчики и преобразователи информации систем измерения, контроля и управления» (Датчик – 99). Гурзуф. 1999.

УДК 519.725

ДЕСКРИПТОРНЫЕ АЛГОРИТМЫ ПРЕОБРАЗОВАНИЯ ФУРЬЕ

Широков О.Ю., Прохоров С. А., Овсянников А. С.

Цифровая обработка с использованием преобразования Фурье лежит в основе спектрального анализа дискретных потоков данных. Под дискретным преобразованием Фурье (ДПФ) понимают прямое преобразование

Фурье над дискретной периодической последовательностью $x(n)$ длиной N отсчетов $/2/$:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp(-j2\pi \frac{n}{N}k) = \sum_{n=0}^{N-1} x(n) W^{nk}, \quad (1)$$

где W^{nk} – весовые коэффициенты преобразования;
 n – номер отсчета входной последовательности;
 k – номер вычисляемой гармоники.

Традиционно, структуры устройств вычисления преобразования Фурье строятся или с использованием памяти весовых коэффициентов W^{nk} , или путем непосредственного их вычисления. При этом задача повышения производительности решается путем учета свойств базисных функций, построением вложенных и гнездовых алгоритмов, а также путем разработки сигнальных процессоров с адаптированным набором команд и структур поточно–конвейерного типа $/3/$. При этом, наряду с достигаемым эффектом, существенно повышается структурная сложность вычислений. Это, в свою очередь, приводит к увеличению затрат на реализацию программной и аппаратной частей устройства.

Известно, что операция умножения требует значительно больших вычислительных затрат, чем операция суммирования. Поэтому все модификации алгоритма ДПФ направлены, в основном, на сокращение числа умножений отсчетов входной последовательности на весовые коэффициенты. Исходя из этого, логично предложить алгоритм, не использующий таких операций вовсе. Это достижимо путем применения современной модификации табличного метода вычислений, при котором заранее просчитаны все возможные варианты этих произведений. Возможность исследований в данном направлении подтверждается постоянным ростом объема и быстродействия микросхем памяти на фоне снижения их стоимости. Тогда, в общем случае, алгоритм примет вид:

$$X(k) = \sum_{n=0}^{N-1} W^{xnk}, \quad (2)$$

где $W^{xnk} = x(n) \exp(-j2\pi \frac{n}{N}k) = Mem[x(n), n, k]$ – элемент трехмерной матрицы.

Для простоты, рассмотрим преобразование Фурье вещественной последовательности данных:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) \left[\cos\left(\frac{2\pi n k}{N}\right) + j \sin\left(\frac{2\pi n k}{N}\right) \right] = \\ &= \sum_{n=0}^{N-1} \left[W_{RE}^{xnk} + j W_{IM}^{xnk} \right]. \end{aligned} \quad (3)$$

Метод можно назвать “дескрипторным”, т.к. ячейка памяти хранит значение произведения базисной функции на выбранный уровень квантования, а адресуется при помощи номера отсчета n , его амплитуды $q=x(n)$ и номера вычисляемой гармоники k . Очевидно, что вычисление одного коэффициента Фурье требует N операций комплексного сложения и не использует умножения. Блок-схему вычислителя можно представить в виде:

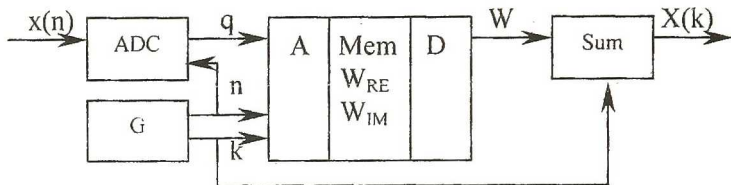


Рисунок 1.

Управляющий генератор G формирует метки времени n и метки частот k , сигнал q на выходе АЦП используется в качестве третьего адреса для матрицы Mem , а значения W накапливаются в соответствующих ячейках сумматора Sum .

При этом единственным критерием эффективности метода является объем необходимой памяти. Так как при анализе процессов с нулевым средним обычно используют АЦП с симметричным относительно нуля динамическим диапазоном $1/4$, это позволяет сократить требуемое количество кодируемых уровней до $2^{(R-1)}$, где R – разрядность входного сигнала. Таким образом, коэффициентов для вычисления $(N/2+1)$ гармоник потребуются:

$$Mem(Re, Im) = 2 * n_{max} * k_{max} * q_{max} = 2 * N(N/2+1) 2^{(R-1)} \approx N^2 2^{(R-1)}. \quad (4)$$

В схеме на рисунке 2 знаковый разряд АЦП используется для изменения полярности при суммировании в накопительных регистрах.

Управление данной системой фактически сведено к генерации тактовых импульсов и своевременному учету знака входного отсчета.

Рассмотрим случай, когда длина входной последовательности N кратна степени двойки ($N=2^M$).

Тогда (4) можно записать: $Mem(Re, Im) \approx 2^{(2M * R - 1)}$. (5)

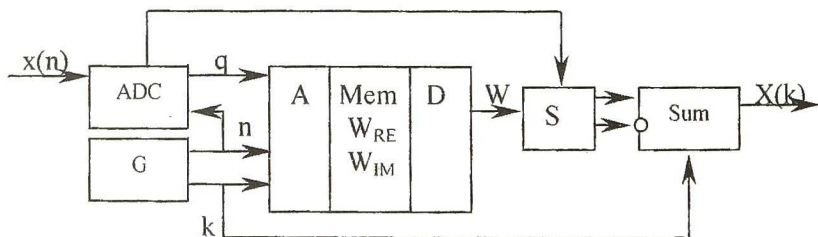


Рисунок 2.

Представленная форма алгоритма является наиболее простой и быстрой, но требует значительных объемов памяти коэффициентов даже при малых длинах входной последовательности.

Рассмотрим модификации алгоритма на основе свойств базисных функций, ведущие к снижению затрат памяти. Так как функция $\cos(2\pi nk/N)$ симметрична относительно $N/2$, а $\sin(2\pi nk/N)$ – относительно $N/4$ и $3N/4$, то имеется возможность в два раза сократить количество хранимых коэффициентов. При этом, однако, нарушается линейная организация памяти коэффициентов, т.е. адрес ячейки не соответствует номеру коэффициента. Это делает невозможным применение индексной адресации.

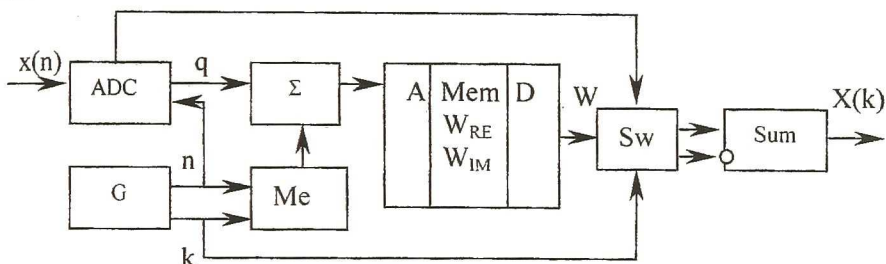


Рисунок 3.

Для разрешения этой проблемы можно применить более сложное устройство управления или, например, схему косвенной адресации на основе таблицы дескрипторов (рисунок 3). Свойства нечетности косинуса и четности синуса позволяют еще вдвое уменьшить количество ячеек с учетом фиксации изменения знака (рисунок 4). Для этого можно использовать, например, знаковый разряд адреса из таблицы дескрипторов.

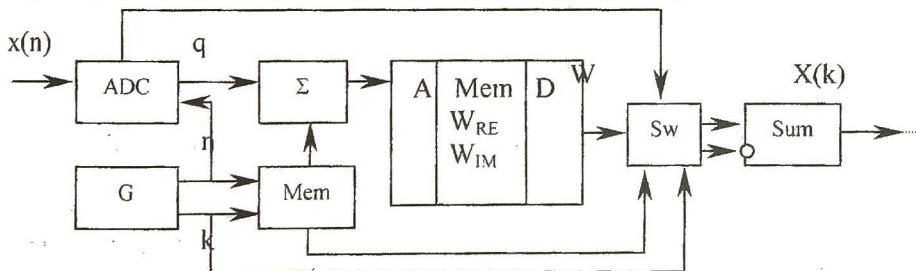


Рисунок 4.

Таким образом, для действительной и мнимой частей преобразования достаточно вычислить значения базисных функций на интервале $(0; \pi/2)$.

В силу ортогональности базисных функций для вычисления действительной и мнимой частей преобразования можно использовать один вдвое

меньший массив памяти коэффициентов. Для этого нужно внести соответствующие изменения в таблицу дескрипторов или схему управления.

В итоге, ввод дополнительной матрицы адресов и схемы управления позволяет сократить матрицу коэффициентов практически на порядок:

$$Mem \approx 2^{(2M \cdot R - 1)}. \quad (6)$$

В то же время, к разрядности таблицы дескрипторов предъявляются значительно менее строгие требования. Еще более значительное сокращение можно произвести исходя из того, что коэффициенты 0, 1 и -1 фактически не требуют размещения в массиве памяти.

Перечисленные приемы лежат в основе всех классических быстрых алгоритмов преобразования Фурье. В общем виде:

$$W^{nk} = W^{(n \cdot N)k}; \quad W^{N \cdot nk} = (W^{(n \cdot N)k})^* \quad (7)$$

Однако в них нужная адресация коэффициентов достигается путем создания вложенных структур, перестановкой и модификацией матриц преобразования.

Следует отметить, что точность предлагаемого метода идентична классической форме ДПФ при условии, что квантование произведений в обоих случаях одинаково. При этом дополнительные затраты, связанные с более высоким объемом памяти, компенсируются повышением быстродействия за счет исключения высокзатратных операций умножения. Сокращение количества тактов, приходящихся на цикл обработки, способствует применению данной схемы для расчетов в условиях реального масштаба времени.

Рассмотрим возможность модификации быстрых алгоритмов преобразования Фурье в соответствии с предлагаемым методом. Такие вычисления обычно реализуются посредством программ для ЭВМ или цифровых сигнальных процессоров. Как правило, эти алгоритмы организованы в виде многоступенчатых процедур вычисления ДПФ длинной последовательности через преобразование более коротких последовательностей $1/1$. Это накладывает определенные ограничения на использование предлагаемого алгоритма. Так, метод Кули-Тьюки разбивает преобразование на несколько этапов, на каждом из которых к отсчетам последовательности применяется преобразование “бабочка”. В общем случае на каждом этапе вычислений происходит умножение сигнала на поворачивающий коэффициент. Это должно неограниченно увеличивать количество кодируемых уровней квантования. На практике, для избежания переполнения сумматоров производится масштабирование промежуточных результатов на входе каждой операции “бабочка” путем сдвига данных вправо на определенное число разрядов. Это ограничивает динамический диапазон,

сокращает число кодируемых уровней и делает возможным замену операции умножения выборкой из памяти. Как и ранее, алгоритм оптимизируется с учетом свойств базисных функций и тривиальности умножений на ± 1 и $\pm j$. Например, известно, что 16-точечный алгоритм БПФ Кули–Тьюки по основанию два требует 152 вещественных сложения и 24 вещественных умножения.

Малые БПФ-алгоритмы Винограда на основе быстрых круговых сверток, реализуются тремя этапами: сложениями, сгруппированными в матрицы предварительных и постсложений, и умножениями на диагональную матрицу коэффициентов. Промежуточные данные при этом также требуют масштабирования, причем его способы могут быть различными для преобразований различной длины. Например, 16-точечное преобразование содержит 10 вещественных умножений и 74 вещественных сложения, а динамический диапазон на входе умножителя, в общем случае, может в 16 раз превышать разброс значений входного сигнала. Способ объединения алгоритмов Винограда малой длины для вычисления преобразования Фурье большой длины будет определять результирующий порядок операций сложения и умножения, а следовательно, и способ масштабирования промежуточных данных. При этом тривиальные умножения в малом алгоритме приводят к нетривиальным умножениям (и сложениям) в алгоритме большой длины.

Общей задачей при реализации предлагаемого алгоритма вычисления ДПФ является достижение оптимального соотношения между объемом необходимой памяти и методическими ошибками преобразования: вносимой ошибкой масштабирования и квантования коэффициентов. Естественно, что выборка из памяти должна быть реализована меньшими затратами по сравнению с операцией умножения. Предлагаемый метод может быть эффективно использован для оценки спектральной функции по коротким реализациям данных, оценки спектральной плотности мощности длинных последовательностей методом “скользящего” БПФ или методом периодограмм, а так же при вычислении отдельных частотных составляющих спектра.

Список использованных источников

1. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов: Пер. с англ. – М.: Мир, 1989. – 448 с.: ил.
2. Гольденберг Л. М. и др. Цифровая обработка сигналов: Учеб. Пособие для вузов / Л. М. Гольденберг, Б. Д. Матюшкин, М. Н. Поляк. – 2-изд., перераб. и доп. – М.: Радио и связь, 1990. – 256 с.: ил.
3. Лосев В. В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: Учеб. Пособие для вузов. – М.: Выш. шк., 1990. – 132 с.: ил.
4. Микросхемы памяти. ЦАП и АЦП: Справочник–2-е изд., стереотип / О. Н. Лебедев, А-Й. К. Марцинкявичюс. Э-А. К. Багданские и др.; – М.: КубК-а, 1996 – 384 с.: ил.